情報保護學會論文誌
第 21 卷 第 5 號, 2011. 10

실용적이고 안전한 RFID 요킹증명 프로토콜*

함 형 민,† 송 주 석‡
연세대학교

Practical and Secure Yoking-Proof Protocol for RFID*

Hyoung-min Ham,† JooSeok Song‡
Yonsei University

요 약

RFID 요킹증명은 2004년 A. Juels에 의해 제안된 개념으로써, 한 쌍의 태그가 하나의 리더에 의해 동시에 스캔됐음을 증명하는 것이다. A. Juels가 최초로 제안한 요킹증명 프로토콜은 이 후 재생공격에 취약한 문제점이 지적되었고, 이를 개선하기 위해 여러 다른 요킹증명 프로토콜들이 제안되었다. 그러나 최초 제안된 기법이 태그의 낮은 성능을 고려해 프로토콜의 경량화를 중요한 이슈로 보았던데 비해, 현재까지 알려진 요킹증명 프로토콜들은 상대적으로 태그에 더 많은 연산량과 저장공간을 필요로 하는 단점이 있다. 우리는 요킹증명의 기본적인 조건을 모두 만족시키면서 성능이 낮은 수동형 태그를 고려한 안전한 프로토콜 두 가지를 제안한다. 제안하는 기법은 기존에 알려진 재생공격 뿐만 아니라 전수공격에 있어서도 높은 안전성을 보장하며, 한 쌍 이상의 태그 그룹, 혹은 다수의 태그 그룹들을 한 번의 프로토콜 수행으로 증명할 수 있도록 설계되었다.

ABSTRACT

Yoking proof is a concept proposed by A. Juels in 2004. It proves that a pair of tags are scanned simultaneously by one reader. After the first yoking proof protocol is proposed by A. Juels, replay attack vulnerabilities of yoking proof are considered and many other yoking proof schemes are proposed to improve it. However, compared with the first yoking proof scheme which emphasizes protocol efficiency due to the limited performance of tags, other yoking proof protocols need more computing power and storage of the tags. We propose two security protocols that consider both the general condition and limited performance of tags. The proposed scheme can protect the tags from replay attack and Brute-force attack as well. Moreover, many pairs of tags or several tag groups can be proved at the same time by executing the protocol only once.

Keywords: RFID, Yoking proof, Grouping proof

I. 서 론

RFID(Radio Frequency IDentification)는 IC칩과 무선을 통해 다양한 개체의 정보를 관리할 수 있는 차세대 인식기술이다. 물류관리부터 유비쿼터스 센서 네트워크 분야에 이르기까지 다양한 응용에 사용될 수 있으며, 특히 바코드를 대체할 기술로 기대 되고 있다. 2004년, A. Juels는 요킹증명이라는, RFID와 관련된 새로운 개념을 제시했다. 요킹증명의 목적은 두 태그가 동시에 존재했음을 증명하는 것이다. 이를 위해 요킹증명 기법은 한 쌍의 태그가 사전에 정의된 프로토콜 수행시간 동안, 하나의 리더에 의해 동시에 스캔되었음을 보장할 수 있는 증명값(proof)을 생성하도록 설계되어야 한다. 요킹증명은

접수일(2010년 12월 14일), 수정일(2011년 5월 23일), 게재확정일(2011년 7월 17일)
* 이 논문은 2010년도 정부(교육과학기술부)의 재원으로 한국연구재단의 기초연구사업 지원을 받아 수행된 것임 (2011-0002806)
† 주저자, hmham@emerald.yonsei.ac.kr
‡ 교신저자, jssong@emerald.yonsei.ac.kr

다양한 응용에 적용할 수 있다. 예를 들면 한 상자 안에 같이 포장되어야 하는 상품들의 누락을 방지할 수 있으며[1], 수술 중 의료기구 들이 유실되지 않도록 검사하는 것도 가능하다[2,3]. 이처럼 요킹증명은 태그가 부착된 객체들 중 일부가 동시에 존재하는지 여부를 확인해야하는 응용에 적용될 수 있다. 그러나 악의적인 공격자는 태그복제, 재전송 등을 통해 한 쌍의 태그 중 어느 한 쪽이 없거나, 혹은 둘 모두가 존재하지 않는 상황에서도 증명이 성립되도록 할 수 있다. 실제로 최초의 요킹증명 기법은 정식으로 프로토콜에 참여하는 리더의 악의적인 행위와 태그의 성능상 한계로 인한 제약을 고려하여 설계되었으나 재전송 공격에 취약한 문제가 있었다. 이 후의 연구들을 통해 재전송 공격 문제는 개선되었으나, 초기의 기법에 비해 상대적으로 높은 수준의 연산과 많은 양의 저장공간이 필요하며, 향상된 태그의 성능을 가정하게 되었다. 최근 들어 RFID 수동형 태그의 연산능력 향상 및 단가하락으로 RFID 태그에 적용될 수 있는 경량화된 암호 알고리즘에 관한 연구가 주목 받고 있으나, 물류관리 시스템 같이 대량의 객체 식별 및 제한된 수명 주기, 그리고 소형화된 태그를 요구하는 응용에는 현실적인 비용을 생각할 때 적합하지 않다. 우리는 최초의 기법과 동일한 조건으로 보다 안전하며 실용적인 요킹증명 프로토콜을 제안한다. 제안 기법은 재생공격에 안전할 뿐만 아니라, 전수공격에도 강건하며, 한 번에 한 쌍 이상의 태그, 혹은 다수의 태그 쌍을 포함하여 증명값을 생성할 수 있다.

본 논문의 구성은 다음과 같다. Ⅱ.관련연구에서는 연구에 필요한 기반지식과 기존의 연구를 소개하고, Ⅲ. 제안기법에서는 기존에 알려진 요킹증명의 요구사항을 설명한 후, 실용성 확대를 위한 추가 요구사항을 정의하고 이를 만족하는 두 가지 프로토콜의 수행 단계 및 특징을 설명한다. 그리고 Ⅳ. 분석에서는 안전성 분석을 위한 공격모델을 정의하고, 제안기법의 재생공격과 전수공격에 대한 강건성을 기존의 기법과 비교분석한다. 또한 경량화 여부를 보이기 위해 프로토콜의 연산량, 요구되는 저장공간, 그리고 요구사항의 만족여부를 기존의 기법과 비교분석한 후, 마지막으로 Ⅴ.결론에서 끝을 맺는다.

Ⅱ. 관련연구

2.1 요킹증명

요킹증명은 태그를 식별하는 최소단위가 하나 이상이며, 프로토콜에 참여하는 리더의 신뢰성 여부와 관계없이, 특정 태그 그룹의 태그들이 검증자가 요청한 시점에 식별됐다는 점을 확인할 수 있게 한다. 요킹증명은 크게 세 가지의 요소로 구성된다. 첫 번째는 검증자, 두 번째는 태그, 그리고 리더이다. 검증자와 리더는 사전에 상호 인증되었으며, 안전한 채널을 사용한다. 그러나 리더는 잠재적 공격자로써, 신뢰할 수 없다고 가정한다. 이 같은 환경에서 요킹증명의 필요성을 다음 예를 통해 설명할 수 있다.

- 검증자는 태그들이 실제로 존재하는지 확인하고자 한다.
- 검증자의 요청을 받은 리더는 태그들을 스캔하고, 그 응답을 검증자에게 전달한다.
- 검증자는 응답을 받아 태그들이 존재하고 있음을 확인한다.

만일 리더가 사전에 스캔해 둔 어느 한 쪽의 태그의 응답(혹은 양 쪽 모두)을 곧바로 검증자에게 전송하지 않고, 검증자로부터 다음 요청이 있을 때 재전송할 경우, 검증자는 이를 구별할 수 없다. 이 같은 리더의 부정을 제한하기 위해 요킹증명 기법에서는 일반적으로 검증자가 프로토콜을 시작하고 프로토콜에 직접 참여한다. 사전에 지정된 한 쌍(혹은 그 이상)의 태그들은 자신들의 응답을 하나로 합쳐서 서로 인접해 있었음을 보이며, 이 때 리더는 단지 태그와 태그 사이를 연결하는 매개의 역할만 하게 된다. 요킹증명 프로토콜을 통해 생성된 증명값은 프로토콜에 참여한 검증자에게만 의미가 있으며, 다른 제 3자에게 제시될 수 없다. 그러나 두 태그가 동일한 세션에 스캔되었는지에 대한 하나의 근거로 활용될 수 있다[1]. 그 외에도 요킹증명 프로토콜은 경량화를 고려해야 한다. 특히 프로토콜에서 요구하는 연산 및 특정 파라메터가 태그에 집중되지 않도록 해야 한다.

2.2 태그의 동시성과 RFID 인증기법

기존의 RFID 인증기법은 태그의 인증 여부 외에 인증된 시점을 고려하지 않으며, 프로토콜에 참여하는 리더와 DB를 신뢰성 있는 하나의 개체로 보고 내부자의 부정을 고려하지 않기 때문에, 위의 2.1에서 언급된 태그의 응답에 대한 동시성 문제를 방지할 수 없다. 만일 리더의 부정을 제한하기 위해 RFID 인증기법을 수정하여 검증자가 프로토콜을 시작하고, 태그의

응답을 수신하기까지 소요되는 시간을 제한해도, 요킹 증명 기법과 동등한 수준의 성능을 보장할 수 없다. RFID 인증기법은 리더의 통신범위 안의 모든 태그를 대상으로 동작하며, 태그의 응답간의 충돌문제를 별도로 고려하지 않기 때문이다. 잦은 태그 응답간의 충돌은 신뢰성 있는 동시성 보장을 어렵게 한다. 이에 비해 요킹증명 기법은 리더가 한 번에 일정한 수의 태그들로 이루어진 태그 그룹을 최소단위로 통신하도록 설계되며, 각 태그는 사전에 부여된 순서를 기반으로 순차적으로 응답한다. 이를 통해 요킹증명 기법은 태그 응답간의 충돌을 최소화하고 보다 신뢰성 있는 동시성 증명을 제공한다.

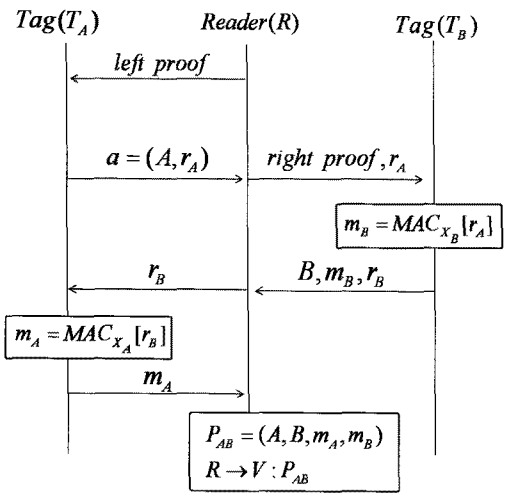
2.3 경량화MAC

A. Juels는 표준 암호 기법을 이용한 Basic yoking proof protocol과 경량화MAC 기법을 이용한 One-time yoking proof protocol 두 가지를 제안하였다. Basic yoking proof protocol은 일방향 해쉬함수와 표준 MAC 함수를 사용하는데 비해, One-time yoking proof protocol은 표준 암호화 함수를 사용할 수 없는 태그를 위해 경량화MAC (Minimalist MAC)기법[1]을 사용하였다. 경량화 MAC은 A. Juels가 표준 암호 알고리즘을 적용할 수 없는 태그에 요킹증명을 적용하기 위해 제안한 경량화된 MAC 함수이다. 경량화MAC은 입력된 비트열을 비밀키의 각 비트와 대응시켜 MAC값을 출력한다. 여기서 비밀키는, 입력 비트열의 각 비트 0과 1에 대응하는 한 쌍의 비트열이다. 입력된 비트열과 동일한 길이를 가지는 두 개의 비트열 쌍이, 입력 비트열의 각 자리 0, 1비트에 대응하여 MAC값을 출력한다. 이 기법은 성능이 낮은 태그를 고려하여 매우 단순한 구조로 안전성을 보장하지만, 1회 MAC을 생성하는데 키 길이의 두 배의 메모리가 필요하며, 무엇보다 사용횟수가 제한적이라는 단점을 가지고 있다. 이 때문에 일정 수준의 인식률을 보장할 수 없는 환경에 적합하지 못하고 서비스 거부 공격(DoS attack)에 취약하므로 다양한 환경에 사용되기는 어렵다. 하지만 경량화MAC은 메모리가 증가되면 저장된 비밀키 수만큼 안전한 MAC값을 생성할 수 있어, 적은 전력으로 동작하는 수동형태그에 적합한 방식이다. 또한 기존에 암호연산에 사용되던 전력을 송신에 사용할 수 있으므로, 태그 안테나의 소형화 측면에서도 유리하다. 이후에 제안된 요킹증명 기법들은 대부분 표준 MAC 함수를 사용하는 태그를 가정하였다[4-8]. 이러한 기법들은 표준 MAC 함수를 통해 요킹증명의 요구조건을 만족하며, 동시에 경량화MAC의 단점을 보완하였으나, 오히려 현실성은 최초의 기법에 비해 떨어진다. 현재 국제표준규격 중에서도 상대적으로 높은 성능을 가지고 있는 UHF방식의 ISO-18000-6[9]이나, EPCglobal class1 gen2[10]규격의 수동형 태그는 MAC연산을 지원하지 않는다[11,12]. 이처럼 현재 태그의 성능과 요킹증명에서 요구하는 성능에 차이가 있는 만큼, 경량화MAC기법은 제한된 연산횟수 문제를 제외하면 유용하게 사용될 수 있으며, 경량화 MAC기법의 제한적인 횟수를 보완하기 위한 연구[13]도 발표되었다.

2.4 요킹증명 기법들

여기서는 이전에 제안되었던 요킹증명 프로토콜을 설명하고, 각 기법의 특징을 분석한다. 단, 요킹증명의 기본적인 요구사항 외에 추가적으로 프라이버시 문제를 고려한 기법은 대상에서 제외하였다. [그림 1]은 A. Juels에 의해 최초로 제안된 요킹증명 프로토콜의 수행과정을 나타낸다. 이 프로토콜에서 한 쌍의 태그 T_A와 T_B는 각각 자신의 유일한 비밀키 X_A, X_B, 고정된 응답순서, 그리고 랜덤 난수를 가지고 있으며, 이를 검증자와 공유하고 있다. 만일 프로토콜 수행시간이 사전에 정의된 시간 Δ를 넘기면 자동으로 중지된다고 가정한다. 프로토콜은 다음과 같은 단계로 이루어진다.

1. 검증자의 요청을 받은 리더는 *left proof* 로 T_A에 요청한다.
2. T_A는 사전에 저장된 난수 r_A와 자신의 이름 A로 응답한다.
3. 리더는 T_B에 대한 요청 *right proof* 와 함께 r_A를 T_B로 전달한다.
4. T_B는 r_A를 이용해 경량화MAC 함수로 m_B를 생성하고, 사전에 저장된 난수 r_B와 자신의 이름 B를 리더에게 보낸다.
5. 리더는 T_B로부터 받은 난수 r_B를 T_A에 전달한다.
6. T_A는 r_B를 이용해 경량화MAC 함수로 m_A를 생성하고, 이를 리더에게 보낸다.
7. 리더는 증명 $P_{AB}=(A, B, m_A, m_B)$를 검증자에게 보낸다.
8. 검증자는 사전에 태그와 공유된 비밀 키와 난수를 이용해 P_{AB}를 검증한다.



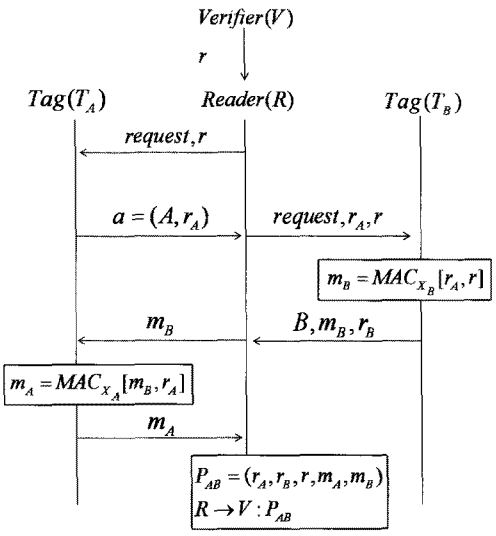
[그림 1] One-time Yoking proof

이 기법의 특징은 태그가 1회성의 난수로 MAC 값을 생성한다는 점이다. 태그는 검증자와 사전에 공유된 난수를 바탕으로 MAC 값을 생성한다. 필요할 경우 MAC생성 횟수를 늘릴 수 있지만, 그만큼 태그에 더 많은 메모리가 요구된다는 단점이 있다. 이 때문에 경량화MAC을 사용할 경우 해당 기법의 검증횟수는 태그의 성능이 낮을수록 제한적이다. 또한 T_B는 *right proof*와 함께 r_A를 받을 때 자신과 한 쌍인 T_A의 메시지를 구별할 수 없으므로, 동시에 다수의 태그 쌍을 검증할 수 없어 비효율적이다. 그 외에도 이 기법은 공격자가 태그 A로부터 사전에 미리 정보를 받아둘 수 있고, 이후 태그 T_A없이 태그 T_B와 사전에 받아둔 정보만으로도 검증자에게 정상적인 증명을 제시할 수 있어 재생공격에도 취약하다.

Saito와 Sakurai는 재생공격이 가능한 문제를 지적하고, 이를 개선하기 위해 타임스탬프를 사용하는 프로토콜[4]을 제안하였다. 또한 shrinkage[1]의 예를 통해, 리더가 잠재적 공격자일 경우 발생할 수 있는 피해와 이를 방지해야 하는 근거를 제시하였으며, Pallet tag라는 고성능의 태그를 포함하여 한 쌍 이상의 태그에 대해서도 요킹증명이 가능한 그루핑 증명(Grooping proof)을 제안하였다. Saito 등의 기법은, 우선 검증자가 난수 대신 타임스탬프 TS를 리더에게 전달하고, 태그가 이 TS를 MAC 생성에 사용하는 것이 특징이다. 하지만 공격자는 이전 세션의 TS를 기준으로 다음 TS의 범위를 예측할 수 있으므로, 사전에 예측 범위내의 TS들을 전송하여 그에 대한 T_A의 응답들을 수집해 둘 수 있다. 그 후 검증자로부터 정상적인 요청이 들어왔을 때 공격자는 사전에 수집해 둔 응답을 이용해 재전송 공격이 가능하다.

S. Piramuthu는 Saito 등의 기법이 재생공격의 문제가 있다고 지적하고, 이를 해결하기 위해 검증자가 프로토콜 수행시간을 모니터링하는데 사용했던 TS를 랜덤 난수로 대체한 프로토콜[5]을 제안하였다. 태그는 검증자가 보낸 랜덤난수 r을 Seed로 사용하여 응답을 생성해내는 방식이다. 그러나 태그의 성능이 높지 않고, 이 때문에 충분히 안전한 길이의 r을 생성하지 못한다면 공격자는 전수공격을 시도할 수 있다. 이 프로토콜은 [그림 2]에 나타나 있다.

2008년 조정식 등은 이 같은 점을 지적하고, 기존과 달리 재전송공격을 방지하는 것이 아닌, 전수공격에 강건한 프로토콜[8]을 제안하였다. 또한 안전성 분석을 위해 전수공격을 수행하는 공격자를 가정하였다. 이들이 제안한 프로토콜은 [그림 3]에 나타나 있다. 이 프로토콜의 특징은 검증자가 요청과 함께 보내는 난수의 숫자를 두 개로 늘려 전수공격에 대한 복잡도를 증가시키고자 한 것이며, 프로토콜의 동작은 S. Piramuthu의 Modified yoking proof 프로토콜을 기본으로 하고 있으므로, 여기서는 별도로 설명하



[그림 2] Modified yoking proof

1) Shrinkage는 재무회계에서 사용되는 용어로 계속기록법에 의해 계산된 장부상의 재고와 실지조사를 통해 파악된 실제재고와의 차이이다. 여기서는 제품이 공급에서 판매되기까지의 과정 중 재고에서 사라져 파악이 불가능한 경우를 말한다.

지 않는다.

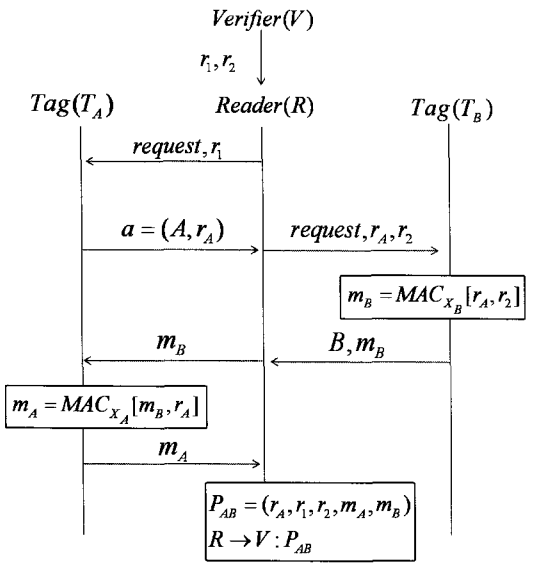
III. 제안기법

제안하는 기법은 2.2에서 언급한 것처럼, 수동형 태그에 적합한 암호학적 함수를 태그의 연산능력이 아닌, 메모리 공간을 활용하는 방식으로 보고 이를 현실적이면서도 요킹증명에 적합한 태그의 성능으로 가정한다. 제안하는 프로토콜은 다음 요구사항을 목표로 한다.

- 요킹증명: 첫 번째 요구사항은 요킹증명에 대한 기본적인 조건을 말한다. 즉, 요킹증명 프로토콜은 한 쌍의 태그가 동시에 스캔되었음을 보장할 수 있어야 한다.
- 그룹요킹증명: 그룹요킹증명은 L. Bolotnyy 등이 A. Juels의 요구사항을 확장한 것으로써, 둘 이상의 태그에 대한 요킹증명을 의미한다[6]. 그룹요킹증명은 Pallet tag와 같이 성능이 뛰어난 별도의 태그가 필요하지 않다는 점에서 Saito와 Sakurai가 제안한 그루핑 증명과는 차이가 있다.
- 안전성: 재전송 공격에 대한 안전성과 전수공격에 대한 강건성을 말한다.
- 멀티증명: 한 번의 프로토콜 수행으로 여러 태그 그룹에 대한 증명을 얻을 수 있는지 의 여부이다. 이 요구사항은 이전에 고려되지 않았던 새로운 요구사항으로, 태그의 모임 T에 속한 사전에 정의된 그룹 T_{sub}의 증명들을 얻기 위해 필요한, 프로토콜의 총 수행 횟수를 감소시킬 수 있다.
- 경량화: 프로토콜 수행 시 태그에 가해지는 부하의 최소화를 의미한다. 여기서는 특히 태그에 요구되는 저장공간과 연산횟수의 최소화를 뜻한다.

우리는 위의 조건을 모두 만족하는 두 가지 프로토콜을 제안한다. 이 프로토콜들은 각각 서로 다른 성능의 태그를 고려하여 설계되었다. 경량화MAC을 사용하는 첫 번째 프로토콜은 관련연구에서 소개한 다른 어떤 기법보다 태그에 주는 부담이 적지만, 경량화 MAC이 가지는 단점 때문에 연산횟수에 제약이 있으며, 응용환경에도 제한이 있다. 두 번째 프로토콜은 표준 MAC 함수를 사용하며 이전의 기법보다 향상된 안전성을 지원한다.

첫 번째 프로토콜은 요킹증명만을 위한 태그를 고려한다. 이 태그는 다양한 크기의 객체에 부착될 수 있도록 충분히 소형화되어 있고, 이로 인해 제한적인 성능을 가지며, 한정된 수명주기를 가지고 있다고 가정한다. 이 때문에 경량화MAC과 같이 사용횟수에 제한이 있는 보안 알고리즘을 지원한다고 가정한다. 반면, 두 번째 프로토콜은 일반적으로 가정하고 있는 태그의 성능을 고려하여 설계되었다. 즉, 기본적으로 태그의 개별적인 식별이 가능하며, 표준암호알고리즘을 사용할 수 있다는 점이 다르다. 그 외에는 기본적으로 첫 번째 태그와 유사한 성능을 가지고 있다고 가정한다. 프로토



[그림 3] Enhanced yoking proof

[표 1] 표기법

기호	의미
V	검증자
R	RFID 리더
T_A, T_B	태그 A, 태그 B
A, B	태그 A, 태그 B의 이름
AB	태그 A, B의 그룹이름
ID_A, ID_B	태그 A, 태그 B의 ID
X_A, X_B	태그 A, 태그 B의 비밀 키
r	랜덤난수
left proof, *right proof*	태그에 보내는 요청 메시지
$MAC(.)$	표준 MAC 함수
$f(.)$	랜덤난수 생성기.
$\|\|$	Concatenate
P_{AB}	태그 A, B의 증명값

콜에 대한 표기는 [표 1]의 표기법을 따른다.

3.1 가정

- 우리는 n개의 태그를 포함하는 시스템을 가정하고, $T=\{t_1, t_2, \ldots, t_n\}$으로 표기한다.
- T는 m개의 태그 그룹 $T_{sub}=\{tg_1, tg_2, \ldots, tg_m\}$로 나뉜다. 이 때 m은 최대 n/2 이다.
- 시스템 초기 설정에서 태그 t_i는 d bit 길이의 유일한 비밀키 x_i를 가지며, 자신과 한 쌍인 다른 한 쪽, 혹은 같은 그룹인 태그들과 같은 그룹이름 tg_i를 공유한다.
- 검증자는 각 태그들에게 사전에 Left Tag, 혹은 Right Tag 중 하나의 역할을 부여한다.
- 검증자는 태그의 모든 비밀키와 그룹이름, 그 밖의 프로토콜 수행에 필요한 모든 파라메터를 공유한다.
- 각 태그 T_i는 표준 MAC 알고리즘이나, 혹은 그와 동등한 안전성을 제공하는 다른 알고리즘을 사용할 수 있다. 여기서 프로토콜에 적용된 일방향 해쉬 함수와 표준 MAC 함수는 각각 수식 (1), (2)와 같이 나타낼 수 있다.

$$f: \{0,1\}^d \times \{0,1\}^* \rightarrow \{0,1\}^d \quad (1)$$

$$MAC: \{0,1\}^d \times \{0,1\}^* \rightarrow \{0,1\}^d \quad (2)$$

만일 MAC 함수와 비밀키 x를 이용해 입력 m의 MAC 값 M을 계산할 경우에는 $M=MAC_x[m]$으로 표기한다. 검증자는 모든 태그 T와 공유하고 있는 비밀키 x_i를 이용해 증명 P를 검증할 수 있다.

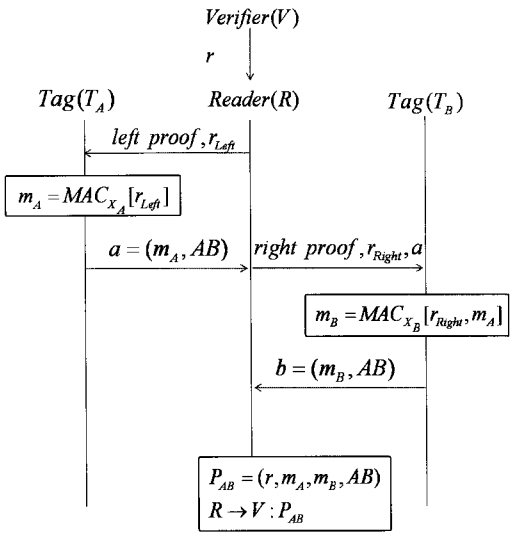
3.2 프로토콜

첫 번째는 경량화MAC처럼 한정된 수명을 가지는 태그를 고려한 프로토콜이다. 이 기법은 관련연구에서 소개된 다른 프로토콜에 비해 태그에게 적은 양의 저장공간과 적은 횟수의 MAC 연산을 요구하며, 재생공격에도 안전하다. 또한 동시에 한 쌍 이상의 태그를 처리할 수 있으며, 무엇보다 이전의 기법들과 달리 한 쌍 이상의 태그 그룹을 동시에 처리하는 것이 가능하다. 다수의 태그 그룹을 한 번에 처리할 수 있는지 여부는 이전에 고려되지 않았던 새로운 요구사항으로, 시스템 T에 속한 모든 그룹 T_{sub}의 증명들을 얻기 위해 필요한, 프로토콜의 총 수행 횟수를 감소시킬 수 있다. 제안하는 프로토콜은 [그림 4]와 같으며, 다음과 같은 단계로 수행된다.

1. 검증자는 요청에 앞서 임의의 난수 $r=(r_{Left}||r_{Right})$를 생성하고, 난수 r을 리더에게 보낸다. 이 때 요청 대상과 r, 그리고 현재 시간을 함께 저장한다.
2. 검증자의 요청을 받은 리더는 *left proof* 와 r_{Left}를 전달한다.
3. Left Tag T_A는 *left proof* 를 받고 경량화 MAC 함수와 비밀키 X_A를 이용해 r_{Left}에 대한 MAC 값 m_A을 생성하고, 이를 태그 쌍의 이름 AB와 함께 응답한다.
4. 리더는 *right proof* 와 r_{Right}, a를 함께 전달한다.
5. Right Tag T_B는 경량화MAC 함수와 비밀키 X_B로 r_{Right}, m_A를 이용해 m_B를 생성하고, 이를 태그 쌍의 이름 AB와 함께 응답한다.
6. 리더는 증명 $P_{AB}=(r, m_A, m_B, AB)$를 검증자에게 보낸다.
7. 검증자는 요청시 저장했던 정보를 이용해 정해진 시간 안에 프로토콜이 수행되었는지 확인하고, 시간이 초과되었을 경우 P_{AB}를 무시한다. 정해진 시간 안에 프로토콜이 수행되었다면 사전에 태그와 공유된 비밀 키와 난수를 이용해 P_{AB}를 검증한다.

이 프로토콜은 다음 요구사항을 만족한다.

- 그룹증명: 이 프로토콜은 5단계에서 T_B의 응답을 받은 후, T_i에게 4, 5단계를 반복적으로 적용



[그림 4] 제안하는 경량화MAC 요킹증명 프로토콜

함으로써, 별도의 수정 없이 해당 프로토콜을 한 쌍 이상의 태그 그룹에 적용할 수 있다. 단, 사전에 태그 그룹은 같은 그룹이름을 공유하고 있어야 하며, 리더의 요청 *left proof, right proof*는 특정 태그의 순서를 가리키는 별도의 요청으로 대체되어야 한다.

- 멀티증명: 해당 프로토콜에서 검증자는 한 번의 요청으로 다수의 태그 쌍, 혹은 그룹에 대한 증명을 받을 수 있다. 이는 관련연구에서 소개된 다른 요킹증명 기법들과 다르게, 각 태그들이 자신의 응답에 자신이 속한 그룹이름 AB를 포함하여, 리더가 한 번에 많은 Left Tag의 응답들을 *right proof* 와 함께 전송해도, Right Tag가 그룹이름을 통해 올바른 요청을 구별할 수 있기 때문이다.
- 경량화: 기존의 기법들은 식별을 위해 태그 ID를 사용하였으나, 요킹증명은 한 그룹단위를 식별대상으로 하므로, 이를 프로토콜에서 제외하고 그룹이름으로 대체하였다. 검증자는 DB에서 태그 ID대신 그룹 ID로 검증대상을 검색한다.

이 기법은 기존 기법에 없던 추가적인 장점에도 불구하고 경량화MAC이 가지는 한계 때문에 다양한 환경에 적용하기는 어렵다. 우리는 이 기법을 기본으로 하여 향상된 안전성을 보장하는 프로토콜을 하나 더 제안한다.

두 번째는 표준 MAC함수를 지원하는 태그를 고려한 프로토콜이다. 일정기간만 사용되고 폐기되거나 초기화되는 태그를 가정한 이전 프로토콜과 달리, 두 번째 프로토콜은 태그가 좀 더 긴 수명을 가지고 있으며, 그만큼 더 많은 위협에 노출될 수 있다고 가정한다. 표준 MAC함수는 태그가 보다 개방적인 환경에 적용되었을 때 상대적으로 높은 안전성을 보장받을 수 있게 해준다. 제안하는 프로토콜은 [그림 5]와 같으며, 다음과 같은 단계로 수행된다.

1. 검증자는 요청에 앞서 임의의 난수 $r=(r_{Left}||r_{Right})$를 생성하고, 난수 r을 리더에게 보낸다. 이 때 요청 대상과 r, 그리고 현재 시간을 함께 저장한다.
2. 검증자의 요청을 받은 리더는 *left proof*와 r_{Left}를 전달한다.
3. Left Tag T_A는 *left proof* 를 받고 경량화 MAC 함수와 비밀키 X_A를 이용해, 자신의 ID와 r_{Left}에 대한 MAC 값 m_A을 생성하고, 이를 태그 쌍의 이름 AB와 함께 응답한다.
4. 리더는 *right proof* 와 r_{Right}, a를 함께 전달한다.
5. Right Tag T_B는 MAC함수와 비밀키 X_B로 자신이 ID와 r_{Right} 그리고 m_A에 대한 MAC 값 m_B를 생성하고, 이를 태그 쌍의 이름 AB와 함께 응답한다.
6. 리더는 증명 $P_{AB}=(r, m_A, m_B, AB)$를 검증자에게 보낸다.
7. 검증자는 요청시 저장했던 정보를 이용해 정해진 시간 안에 프로토콜이 수행되었는지 확인하고, 시간이 초과되었을 경우 P_{AB}를 무시한다. 정해진 시간 안에 프로토콜이 수행되었다면 사전에 태그와 공유된 비밀 키와 난수를 이용해 P_{AB}를 검증한다.

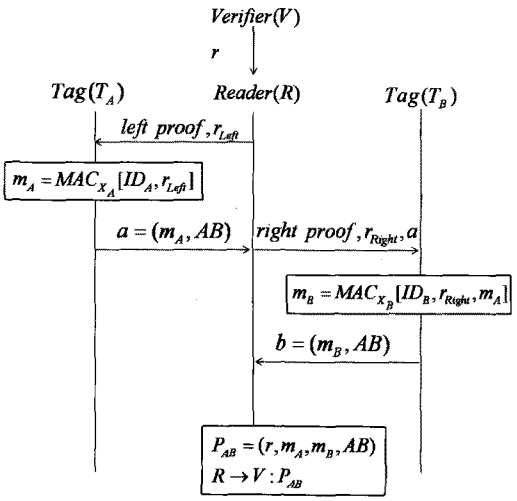
이 프로토콜은 다음과 같은 요구사항을 만족한다.

- 그룹증명: 첫 번째 기법과 동일하다. 단, 이 기법은 태그의 ID를 남겨두어 필요할 경우, 태그를 단독으로 검색하는 기법을 적용할 수 있다. 증명값 생성 과정 중에 일부 태그의 응답이 포함되지 않은 경우, 다시 전체 그룹을 대상으로 프로토콜을 수행할 필요 없이 태그의 ID로 직접 요청하여 검색의 형태로 태그의 존재를 확인할 수 있다.
- 멀티증명: 첫 번째로 제안한 프로토콜과 동일하다.
- 경량화: 앞서 제안한 프로토콜과 달리, 여기서는 태그의 ID를 남겨두었다. 제안 프로토콜에서 태그 ID는 두 가지 기능을 지원한다.

(1) 태그 ID를 평문을 통해 전송하던 기존의 기법들과 달리, 제안 기법에서는 태그 ID를 공개하지 않고, 이를 공격 비용에 추가되도록 하였다.

(2) 한 그룹의 태그 수가 많을 때, 일부 태그의 응답이 누락될 경우, 어떤 태그가 누락됐는지 알 수 없다. 이 경우, 첫 번째 제안과 달리, 여기서는 특정 태그의 ID를 이용해 특정 태그의 응답만을 요구할 수 있다. 그 방법은 다음과 같다.

- 검증자는 실패한 증명 P를 검증하는 과정에서, 어떤 그룹 내의 특정 태그 t_i가 누락됐음을 알고, 이를 확인하기 위해 해당 태그만을 위한 질의를 생성한다. 질의는 $order_i$, r_i, $MAC_{xi}(ID_i, r)$의 형태로 t_i에게 전달된다. 여기서 $order_i$는 요킹증명 시스템이 그룹증명 시스템으로 확장되었을 때, 사전에 정의된 태그의 응답순서를 각 태그에 알리는 역할을 하며, *left proof, right proof*



〔그림 5〕 제안하는 표준 MAC 요킹증명 프로토콜

를 대체한다.

- 검색 질의의 형태는 그룹요킹증명 프로토콜 수행 중에 태그가 수신하는 형태와 동일하므로, 시스템에서는 별도의 수정 없이 해당 기능을 지원할 수 있다.

그룹요킹증명으로 확장이 가능한 점, 멀티증명을 지원하는 점은 제안하는 두 기법 모두 동일하다. 표준 MAC 요킹증명 프로토콜은 첫 번째 프로토콜과 동일한 장점을 가지며, DoS공격에 보다 강건하다. 그러나 태그의 수명이 길어지면서, 전수공격과 같이 긴 공격시간을 필요로 하는 다른 공격에 노출될 가능성을 고려해야한다. 이 경우 태그에 탑재된 MAC함수의 안전성이 프로토콜의 안전성에 직접적인 영향을 미치게 되는데, 기술적 한계, 혹은 비용상의 문제로 인해 MAC 함수의 안전성이 충분히 지원되지 못할 가능성이 있다. 이 때문에 표준 MAC 요킹증명 프로토콜은 안전성을 높이기 위해 전수공격에 대한 복잡도를 고려하여 설계되었다. 전체적인 프로토콜의 흐름은 첫 번째 프로토콜과 유사하지만, 각 태그가 MAC 값을 생성할 때 이전의 기법과 달리 자신의 ID를 포함하며, 이는 결과적으로 전수공격에 필요한 복잡도를 증가시키게 된다.

IV. 분석

4.1 공격모델

공격자의 목적은 태그가 생성하는 증명을 위조하여 이를 이용해 검증에 성공하는 것이다. 공격자는 다음과 같은 능력을 가지고 있다고 가정한다.

- 도청이 가능하다.
- 원하는 태그의 응답을 얻을 수 있다.
- 같은 알고리즘을 사용한다.
- 충분한 계산능력과 저장공간을 갖는다.
- 태그에 대한 물리적인 공격은 고려하지 않는다.

공격자는 목적을 달성하기 위해 위의 능력을 활용하여 재생공격과 전수공격을 행한다고 가정한다.

4.2 재생공격에 대한 안전성

여기서 사용하는 안전성 분석은 A. Juels의 논문에서 소개된 방식[1]을 기본으로 한다.

- 주장: 랜덤 오라클 가정 하에, 태그가 d bit 길이의 출력을 가지는 MAC 알고리즘을 사용하여 응답할 때, 공격자가 태그의 다음 응답과 동일한 응답으로 증명에 성공할 확률은 약 2^{-d} 이다.

  Case 1: 검증자가 다음 요청에 사용할 랜덤난수 r을 공격자가 사전에 임의로 선택하여 태그들에게 먼저 요청해 두는 경우, 공격이 성공할 확률은 약 2^{-2d}이다.

  Case 2: 검증자가 보낸 요청 r_{Left}에 대해 공격자가 T_A의 응답을 임의로 선택할 경우, 공격이 성공할 확률은 약 2^{-d}이다.

  Case 3: 검증자가 보낸 요청 r_{Right}와 T_A의 응답 a에 대해 공격자가 T_B의 응답을 임의로 선택할 경우, 공격이 성공할 확률은 약 2^{-d}이다.

제안된 프로토콜에서 재생공격을 위해 사전에 수집해둔 어떤 응답도 올바른 검증을 통과할 수 없다. 또한 d의 범위가 충분히 클 경우, 사전 예측에 의존한 공격은 거의 불가능하다. 또한 공격자가 예측을 통해 올바른 값을 선택하는데 도움을 줄 수 있는 어떤 정보도 프로토콜 내에서 수집할 수 없다.

4.3 전수공격에 대한 안전성

MAC 값이 길이가 충분하지 못하다면 공격자는 태그의 비밀값을 알아내기 위해 전수공격을 시도할 수 있다. 이 공격은 최초 한 번의 도청이 성공한 이후 부터는 오프라인으로 진행할 수 있으며, 만일 공격자가 태그의 비밀값을 알아낸다면 검증은 실패하게 된다.

공격자는 다음과 같은 단계로 공격을 수행한다.

1. 공격자는 비밀값을 포함하고 있는 어떤 메시지를 선택하고, 이를 수집한다. 이 때, 이 메시지를 생성하는데 사용된 값들을 함께 수집한다.
2. 동일한 메시지를 생성하기 위해 필요한 비밀값을 랜덤하게 선택하고, 수집한 메시지와 동일한 생성단계를 수행한 후, 같은 메시지가 생성되는지 확인한다.
3. 동일한 메시지가 생성될 때까지 2단계의 예측과 생성을 반복한다.

만일 공격자가 d 비트 길이의 비밀값을 위와 같은 방법으로 알아내려고 한다면, 최대 $2^d * d$ 비트의 저장 공간이 필요하게 된다. 우리는 공격에 필요한 최대 저장량을 공격에 요구되는 최대 비용으로 보고, 이를 기준으로 각 프로토콜의 안전성을 비교한다. [표 2]는 각 기법별로 전수공격에 요구되는 저장비용을 나타낸 것이다.

[표 2]에서의 저장비용은 공격자가 MAC값에 포함되는 비밀값을 알아내기 위해 전수공격을 시도할 경우 소모되는 최대 저장비용을 나타낸 것이다. 각 변수별 공격에 요구되는 비용은 (*MAC 값의 모든 경우의 수 + 공격에 필요한 메시지들*)*데이터 길이* 로 나타낸다. 이 분석의 목적은 각 프로토콜 간에 상대적인 안전성을 비교하는 것이며 편의상 TS를 제외한 나머지 값들은 같은 길이를 가지고 있다고 가정한다.

조정식 등의 기법은 다른 기법들과 달리 전수공격에 대한 안전성을 고려하여 설계되었으나, 본 논문에서 제안된 전수공격모델에서는 기존의 기법들과 동등한 수준의 안전성만을 보장할 수 있으며, 제안된 기법이 전수공격에 대해 가장 높은 안전성을 보임을 알 수 있다.

4.4 효율성

여기서는 프로토콜 수행을 위해 태그에 요구되는 계산량과 저장공간을 비교 분석한다. 우선 태그에 요구되

[표 2] 프로토콜 별 전수공격에 필요한 최대저장비용

프로토콜	최대저장비용				
	X_A	X_B	ID_A	ID_B	합계
Yoking proof	(2^d)d+3d (from m_A)	(2^d)d+3d (from m_B)			$(2^{d+1}+6)$d
Yoking proof using timestamp	(2^e)e+3d (from m_A)	(2^e)e+3d (from m_B)			(2^{e+1})e+6d
Modified yoking proof	(2^d)d+3d (from m_A)	(2^d)d+3d (from m_B)			$(2^{d+1}+6)$d
Enhenced yoking proof	(2^d)d+3d (from m_A)	(2^d)d+3d (from m_B)			$(2^{d+1}+6)$d
제안기법 1 (Minimalist MAC)	(2^d)d+2d (from m_A)	(2^d)d+3d (from m_b)			$(2^{d+1}+5)$d
제안기법 2 (Standard MAC)	(2^{d+1})d+2d (from m_A)	(2^{d+1})d+3d (from m_B)	(2^{d+1})d+2d (from m_A)	(2^{d+1})d+3d (from m_B)	$(2^{d+3}+10)$d

※ d: 태그 변수의 길이, e: TS의 길이

[표 3] 프로토콜 별 층 계산량

	Yoking proof	Timestamp	Modified	Enhanced	제안기법1 (Minimalist MAC)	제안기법2 (Standard MAC)
1st flow		1MAC	1f	1f		
2nd flow	1MAC	1MAC	1MAC	1MAC	1MAC	1MAC
3rd flow	1MAC		1MAC	1MAC	1MAC	1MAC
Total	2MAC	2MAC	1f+2MAC	1f+2MAC	2MAC	2MAC

※ f : 일방향 해쉬 연산, MAC : MAC 연산

〔표 4〕 태그의 패러매터

	Yoking proof	Timestamp	Modified	Enhanced	제안기법1 (Minimalist MAC)	제안기법2 (Standard MAC)
랜덤난수 r	1d*n		1d	1d		
타임스탬프 TS		1e				
비밀키 X	2d*n	1d	1d	1d	2d*n	1d
태그 ID	1d	1d	1d	1d		1d
그룹이름					1d	1d
합계	(3d*n)+1d	2d+1e	3d	3d	(2d*n)+1d	3d

※ d: TS를 제외한 나머지 패러매터 값의 길이, e: TS의 길이

〔표 5〕 기타 요구사항 만족 여부

	Yoking proof	Time stamp	Modified yoking	Enhanced yoking	제안기법1 (Minimalist MAC)	제안기법2 (Standard MAC)
재생공격	가능	가능	안전	안전	안전	안전
자원소모	취약	안전	안전	안전	취약	안전
그룹증명	불가	프로토콜 확장으로 지원 가능	불가	프로토콜 확장으로 지원 가능	지원	지원
멀티증명	불가	불가	불가	불가	지원	지원

는 계산량은 프로토콜이 1회 수행될 때를 기준으로 태그가 수행하는 난수생성과 MAC연산의 총 횟수이며, 〔표 3〕은 각 프로토콜별 계산량을 비교분석한 결과이다. 〔표 4〕는 각각의 프로토콜에서 태그에 요구되는 저장공간을 비교하기 위해, 태그에 저장되어야 하는 패러매터 값의 개수를 비교한 것이다. 편의상 태그의 패러매터들은 타임스탬프 TS를 제외하고 모두 같은 d bits의 길이를 가지고 있다고 가정하고, 랜덤난수처럼 프로토콜 수행 시 매번 생성되는 값은 한 개의 난수를 저장한 것으로 보았다. 그 외에 실제 구현 시 고려되어야 하는 세부 사항은 분석에서 제외하였다.

4.5 실용성

〔표 5〕는 그 외의 요구사항 만족 여부에 대한 비교분석결과이다. 각 기법들의 재생공격에 대한 안전성과 그룹증명과 멀티증명 지원 여부를 나타낸다.

4.1부터 4.5까지의 분석결과들을 종합해 볼 때, 제안된 두 가지 프로토콜은 가장 적은 수준의 연산량과 저장량을 요구하면서도 재생공격, 전수공격에 강건하며, 새롭게 제안된 추가적인 요구사항을 모두 만족함을 알 수 있다.

V. 결론

최초의 요킹증명 프로토콜 이 후의 기법들은 기본적인 요킹증명의 요구사항과는 별개로, 기존에 없던 일방향 해쉬 함수나, 표준 MAC 함수의 사용을 가정하는 등 최초의 기법에 비해 현실성이 떨어지는 단점이 있었다. 이에 우리는 기존의 기법들이 경량화MAC 대신 표준 MAC 함수를 사용해야 했던 원인에 주목하고, 두 가지 MAC 함수의 장단점을 고려한 두 가지의 프로토콜을 제안하였다. 첫 번째 프로토콜은 최초의 기법과 동일하게 경량화MAC을 사용하면서도 요킹증명의 요구조건을 만족하며, 두 번째 프로토콜은 인증횟수에 제약이 있는 첫 번째 기법의 단점을 보완하기 위해 표준 MAC 함수를 사용하는 대신, 전수공격에 강건하도록 설계되었다. 또한 제안된 기법들은 최초의 기법보다 적은 연산량과 작은 저장공간을 태그에 요구하며, 프로토콜에 별도의 추가적인 가정이나 수정 없이 그룹증명과 멀티증명이 가능하다.

참고문헌

[1] Ari Juels, "Yoking-Proofs" for RFID Tags," Second IEEE Annual Conference on Pervasive Computing and Communications Workshops(percomw), pp. 138-143, March 2004.

[2] Hsieh-Hong Huang and Cheng-Yuan Ku, "A RFID Grouping Proof Protocol for Medication Safety of Inpatient," Journal of Medical Systems, Volume 33, Number 6, pp. 467-474, Sept. 2008.

[3] L. Bolotnyy and G. Robins, "Multi-tag RFID systems," Int. J. Internet Protoc. Technol. Volume 2, 3/4, pp. 218-231, Dec. 2007.

[4] Junichiro Saito and Kouichi Sakurai, "Grouping Proof for RFID Tags," In Proceedings of the 19th International Conference on Advanced Information Networking and Applications, pp. 621-624, March 2005.

[5] S. Piramuthu, "On Existence Proofs for Multiple RFID Tags," In Proceedings of the 2006 ACS/IEEE International Conference on Pervasive Services(PERSER '06), pp 317-320, June 2006.

[6] L. Bolotnyy, and G. Robins, "Generalized "Yoking-Proofs" for a Group of RFID Tags," Mobile and Ubiquitous Systems: Networking & Services, 2006 Third Annual International Conference on, pp. 1-4, July 2006.

[7] Pedro P. Lopez, Julio C. Hernandez-Castro, Juan M. Estevez-Tapiador, and A. Ribagorda, "Solving the Simultaneous Scanning Problem Anonymously: Clumping Proofs for RFID Tags," Security, Privacy and Trust in Pervasive and Ubiquitous Computing, Third International Workshop on, pp. 55-60, July 2007.

[8] Jung-Sik Cho, Sang-Soo Yeo, Suchul Hwang, Sang-Yong Rhee, and Sung Kwon Kim, "Enhanced Yoking Proof Protocols for RFID Tags and Tag Groups," In Proceedings of the 22nd International Conference on Advanced Information Networking and Applications - Workshops (AINAW '08), 1591-1596, March 2008.

[9] ISO/IEC 18000-6 Amd 1, "Information technology - Radio frequency identification for item management -Part 6: Parameters for air interface communications at 860 MHz to 960 MHz, Amendment 1 (2006-06-15): Extension with Type C and update of Types A and B," July 2006.

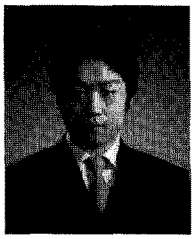
[10] EPCglobal Inc, "Class 1 Generation 2 UHF Air Interface Protocol Standard," Version 1.20, Oct. 2008.

[11] 양연형, 김선영, 이필중, "개선된 수동형 RFID 보안태그와 리더의 인증 및 데이터 보호 프로토콜," 정보보호학회논문지, 20(1), pp. 85-94, 2010년 2월.

[12] Hung-Yu Chien, Che-Hao Chen, "Mutual authentication protocol for RFID conforming to EPC Class 1 Generation 2 standards," Comput. Stand. Interfaces, volume 29, pp. 254-259, February 2007.

[13] 조창현, 이재식, 김재우, 전문석, "경량화된 MAC을 이용한 강력한 Yoking-Proof 프로토콜", 정보보호학회논문지, 19(6), pp. 83-92, 2009년 12월.

〈著者紹介〉



함 형 민 (Hyoung-min Ham) 정회원
2007년 2월: 배재대학교 컴퓨터공학과 졸업
2009년 2월: 한양대학교 컴퓨터공학과 석사
2009년 3월~현재: 연세대학교 컴퓨터과학과 박사과정
〈관심분야〉 RFID, 보안프로토콜, 네트워크보안

송 주 석 (JooSeok Hong) 종신회원
1976년 2월: 서울대학교 전기공학과 졸업
1979년 2월: 한국과학기술원 전기전자공학과 석사
1988년 2월: University of California at Berkeley 컴퓨터과학과 박사
1988년~ 1989년: 미국 Naval Postgraduate School 조교수
1989년 3월~현재: 연세대학교 컴퓨터과학과 정교수
2006년 한국정보보호학회 회장 역임
〈관심분야〉 정보보호, 유무선통신